NuRadioReco

An Event Reconstruction Framework for Radio Neutrino Detectors

Christoph Welling

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

DESY.

Goals

Why another Software Framework?

Flexible

- Usable by any ARIANNA/ARA/RNO-like experiment
- Usable by air shower experiments
- Handle neutrinos + cosmic rays

Modular + Easy to use

- Split tasks into dedicated modules
- Intuitive data structure
- ‘Quality of life’ improvements

Data Structure

Keeping Stuff organized

Detector Description

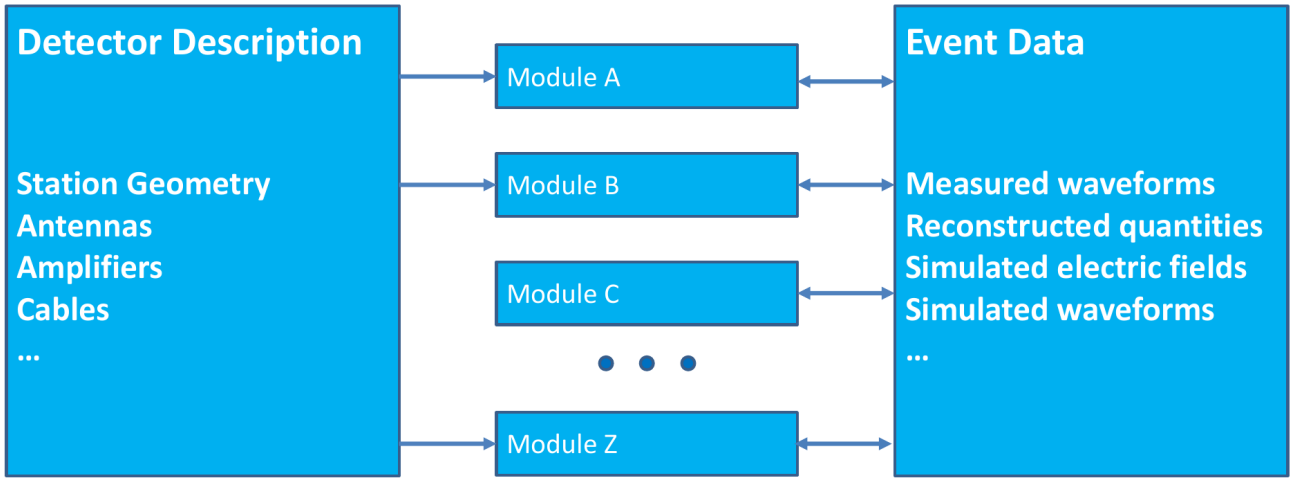
- Provides information on hardware Components
- Does not change during reconstruction

Event Data

- Stores data about the event

Modules

- Execute reconstruction tasks
- Can read Event Data + Detector Description
- Manipulate Event Data, but not Detector description



Detector Description

Antennas, Amps and other Things

Central Database

- Central MySQL database
- Supports time-dependent detector description

JSON Detector Description

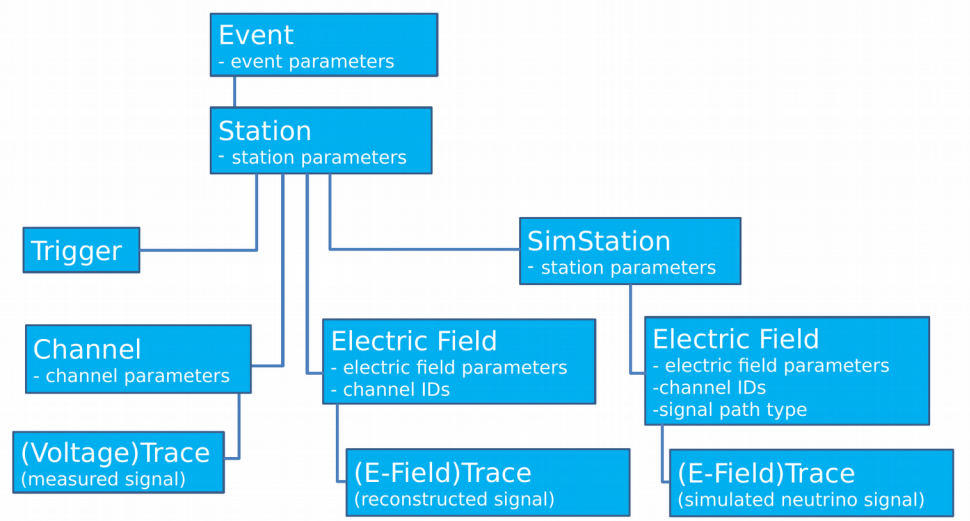
- Detector description in local JSON file
- Generated from MySQL database
- Can be customized (e.g. to test new detector design)
- WIP: Store detector info in event files

Event Data

.nur ein Dateiformat

Own Data Structure

- Input + Output of (almost) all modules
- Supports both Neutrino and CR events
- Can store any stage of event reconstruction
- Can be deserialized in .pickle-like files



Parameter Storage

- (Sim-)Station, Channel, E-field object can store parameters
- Implemented as python Enum

```
import NuRadioReco.parameters.stationParameters as stnp

zenith = station.get_parameter(stnp.zenith)
azimuth = station[stnp.azimuth]
```

Modules

Where Stuff gets done

Dedicated module for each Task

- Independent of each other
- Supports complex if/for logic in reconstruction
- Event can be saved at any point of reconstruction

How to use

- .begin() → Setup before running
- .run() → Execute task
- .end() → Run after task is done

```
from NuRadioReco import eventReader
...
from NuRadioReco import directionFitter

reader = eventReader.event_reader()
reader.begin('filename.nur')
...
direction=directionFitter.directionFitter()

for event in reader.run():
        station = event.get_station(32)
        direction.run(event, station)
        if station.get_parameter(stnp.zenith)<.2:
                writer.run(event)
        else:
                voltageToEfieldConverter.run(event, station)
                writer.run(event)

reader.end()
writer.end()
```

Quality of Life Improvements

Life is already hard enough

Utility functions

- Package of often-used functions (FFT, geometry)

Unit System

- Keeps track of which units properties are given in

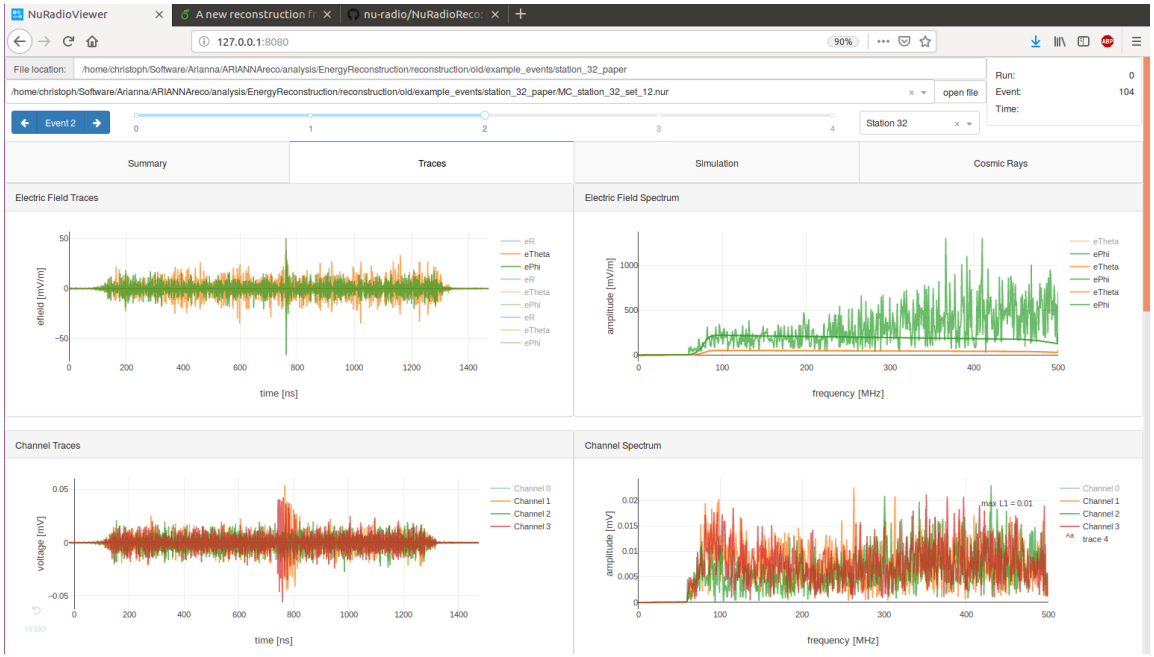
NuRadioViewer

- EventDisplay for .nur files
- Uses plotly/dash
- Rendered by web browser
- Can be deployed online

```
from NuRadioReco.utilities import units

h_everest=8.8*units.km
print('Mt. Everest is %f meter high'%(h_everest/units.m))

plt.plot(channel.get_times()/units.ns,
channel.get_trace()/units.milli/units.Volt)
```



What's left to do?

Want to join?

Use NuRadioReco

- Available with pip:

  pip install NuRadioReco

- Available on github:

  github.com/nu-radio/NuRadioReco

- More information: arxiv: 1903.07023

Help with Development

- New reconstruction modules
- Ongoing development (e.g. coordinate system)